

IN THE CLAIMS:

Please amend the claims as follows:

Claim 34, line 1, please delete "34" and insert --33-- therefor. ✓

REMARKS

Claims 1, 2, 4-7, 10-14, 16, 18-20, 22, and 24-47 are pending in the present application. Claim 34 is amended to overcome a minor informality. Reconsideration of the claims is respectfully requested.

I. Claim Objection

The Office Action objects to claim 34 because it depends on itself. Claim 34, as amended, now depends on claim 33. Therefore, the objection is overcome.

II. 35 U.S.C. § 102, Anticipation

The Office Action rejects claims 1, 2, 5, 7, 10, 13, 14, 16, 18-20, 22, 24-35, 37, 40, 42, and 46 under 35 U.S.C. § 102 as being anticipated by Maruyama et al. (US Patent No. 5,710,920), hereinafter referred to as "*Maruyama*." This rejection is respectfully traversed.

Maruyama discloses a method for extending objects in an object-oriented database. The system and method of *Maruyama* permit an object to be changed in terms of attribute, relation, and procedure independently of schema definition information in the object-oriented database. In contradistinction, the present invention concerns separating the meta data from application code within a distributed data processing system.

In accordance with the present invention, a client is used to input attributes for an object without the software in the client being dependent on the meta data for the object. The software in the client that assists the user in creating a data object is not programmed with the meta data that defines the structure of the object. Instead, the client software must receive the meta data or meta definition from a Meta Data Service in order to, for example, generate graphical user interface fields. This aspect of the invention is covered by claims 1, 2, 4-6, 13, 14, 19, 20, 25, 26, 30, and 31.

Also, in accordance with the present invention, a server in a data processing system may receive data objects in a “soft” format. This soft format comprises a data value stream without meta data that defines the structure of the object. Applications in the data processing system are designed in such a manner that they are not presumed to understand the definition of the data object. Thus, a server must query a Meta Data Service for a meta definition. This aspect of the invention is covered by claims 35-47.

Further, in accordance with the present invention, a Persistent Object Service also receives objects in a soft format. The software in the Persistent Object Service also is not presumed to understand the definition of the data object. Therefore, the Persistent Object Service must query the Meta Data Service for a meta definition and map attributes in the received data streams with attributes in the meta definition of the data object before storing the object to persistent storage. This aspect of the invention is covered by claims 7, 10-12, 16, 18, 22, 24, 27-29, and 32-34.

With respect to claims 1, 13, 19, and 42, the Office Action states:

As per independent claims 1, 13, 19, 42:

Maruyama teaches a method in a software component for processing a data object in a data processing system, the method comprising the computer-implemented steps of:

- sending a query for a meta definition of a data object [e.g., step 1001, fig. 13, col. 9, lines 37-45],
- receiving the meta definition of the data object [e.g., col. 9, line 45].
- identifying object attributes in the meta definition [e.g., col. 9, lines 45-64], line].
- prompting a user to input data values corresponding to the object attributes [e.g., col. 9, line 38].

Office Action dated 14 December 2000. Applicant respectfully disagrees. *Maruyama* teaches a dictionary 108 storing type definition information **within an object-oriented database**. However, *Maruyama* does not teach “identifying attributes in the meta definition” and “prompting a user to input data values corresponding to **the object attributes**” [emphasis added], as recited in claims 1, 13, and 19.

The cited text of *Maruyama* is reproduced as follows:

FIG. 13 is a flow for processing of reflecting a change in type definition when restructuring the DB. The user designates an object identifier of a type definition

object to the object manger 103 through the view manager 101. In FIG. 13, steps 1001, 1004 and 1008 are processed by the dictionary manager 104 and the other steps are processed by the object manager 103.

In step 1001, the type definition object is acquired. The following processing is carried out until information of parts attribute object is determined to be absent in step 1002. In step 1003, it is examined whether the processing for all change information pieces owned by the parts attribute object ends. If unfinished, attribute change information is acquired from the parts object manager 102 in step 1004 and on the basis of this information, attribute information owned by the type definition object of schema is changed. In step 1005, all objects belonging to that type are changed on the basis of the type information after change. At that time, attribute data owned by a structure change object is used. In step 1006, it is examined whether a parts procedure object having procedure change information is present. If present, it is examined in step 1007 whether the processing for all change information pieces owned by the parts procedure object ends. If absent, a parts procedure object is acquired from the parts object manager 102 and on the basis of this information, procedure information of the type definition object in the schema is changed in step 1008.

Maruyama, col. 9, lines 37-64. According to the cited portion of the reference, “[t]he user designates an object identifier of a type definition object.” However, *Maruyama* does not teach or suggest identifying object attributes in the meta definition and prompting the user to input data values corresponding to the **identified** object attributes, as recited in claims, 1, 13, and 19. Each and every claim element is not taught by the prior art; therefore, the claimed invention is not anticipated by the reference.

Present claim 42 recites:

42. A data processing system for processing a data object, said data processing system comprising:
first receipt means for receiving a data value stream for a data object;
sender means for sending a query for a meta definition of a data object;
second receipt means for receiving the meta definition of the data object; and

process means for processing the data object according to attributes in the meta definition for the data object to form a second data value stream.

Maruyama does not teach or suggest “first receipt means for receiving a data value stream for a data object” or “process means for processing the data object according to attributes in the meta definition for the data object to form a second data value stream,” as recited in claim 42. As discussed above, *Maruyama* is concerned only with an object-oriented database and not the manner in which objects are processed in a distributed data processing system. Therefore, *Maruyama* does not teach or suggest the data processing system recited in claim 42 in which an application receives an object in a soft format (data value stream) and queries a Meta Data Service for the meta definition. The Office Action does not address these features; therefore, the rejection must be withdrawn.

With respect to claims 7, 16, 22, 35, and 46, the Office Action states:

As per independent claims 7, 16, 22, 35, 46:

Maruyama teaches a method in a software component for processing a data object in a data processing system, the method comprising the computer-implemented steps of:

- receiving a data value stream [e.g., col. 9, also figs. 12 & 13].
- sending a query for a meta definition of a data object [e.g., step 1001, fig. 13, col. 9, lines 37-45],
- receiving the meta definition of the data object [e.g., col. 9, line 45].
- mapping data values to a data structure according to the attributes in the meta definition of the data object [e.g., col. 9, lines 45-64].

Office Action dated 14 December 2000. Applicant respectfully disagrees. Again, *Maruyama* is concerned only with an object-oriented database and not the manner in which objects are processed in a distributed data processing system. Therefore, *Maruyama* does not teach or suggest the data processing system recited in claims 7, 16, and 22, in which an application receives an object in a soft format (data value stream) and queries a Meta Data Service for the meta definition. The Office Action refers to an entire column of the reference and two flowcharts; however, it is unclear where the step of receiving a data value stream is taught. Furthermore, the cited portion of the reference does not disclose mapping data values to a data structure according to attributes in a received meta definition. *Maruyama* only teaches managing changes to data type

definitions within an object-oriented database. *Maruyama* does not teach that the database receives objects in a “soft” format, i.e. in a data value stream. Each and every claim element is not taught by the prior art; therefore, the claimed invention is not anticipated by the reference.

Claims 35 and 46 recite:

35. A method in a software component for processing a data object in a data processing system, said method comprising the computer-implemented steps of:

- receiving a first data value stream for a data object;
- sending a query for a meta definition of the data object;
- receiving a meta definition of the data object; and
- processing the data object according to attributes in the meta definition of the data object to form a second data value stream for the data object.

46. A computer program product for use with a data processing system for processing a data object, said computer program product comprising:

- a computer readable medium;
- first instructions for receiving a first data value stream for a data object;
- second instructions for sending a query for a meta definition of the data object;
- third instructions for receiving a meta definition of the data object;
- and
- fourth instructions for processing the data object according to attributes in the meta definition of the data object to form a second data value stream for the data object.

Maruyama does not teach or suggest “receiving a data value stream for a data object” or “processing the data object according to attributes in the meta definition for the data object to form a second data value stream,” as recited in claims 35 and 46. As discussed above, *Maruyama* is concerned only with an object-oriented database and not the manner in which objects are processed in a distributed data processing system. Therefore, *Maruyama* does not teach or suggest the method and computer program product recited in claims 35 and 46 in which a software component receives an object in a soft format (data value stream) and queries a Meta Data Service for the meta definition. The Office Action does not address these features; therefore, the rejection must be withdrawn.

Since claims 2, 5, 10, 14, 18, 20, 24-34, 37, and 40 depend from claims 1, 7, 13, 16, 19, 22, and 35, the same distinctions between *Maruyama* and the claimed invention in

claim 1, 7, 13, 16, 19, 22, and 35 for these claims. Additionally, claims 2, 5, 10, 14, 18, 20, 24-34, 37, and 40 claim other additional combinations of features not suggested by the reference.

Particularly, with respect to claim 26, the Office Action states:

As per claim 26:

Maruyama teaches the step of prompting the user for data values comprises:

matching the meta definition to graphical user interface fields; and
presenting the graphical user interface fields to the user [e.g., col. 7, lines 60-65, col. 9, line 5, line 39].

Office Action dated 14 December 2000. Applicant respectfully disagrees. The cited portion of *Maruyama* states:

FIG. 10 is a flow for processing of attribute change. When requesting the present processing, the user transfers an object identifier, a version number and a change request code (appending, deletion and update) upon definition change and change attribute information including attribute name, attribute type, type size and attribute value to the object manager **103** through the view manager **101**.

Maruyama, col. 7, lines 60-66.

When requesting the present processing, the user designates an object identifier, a change request code (appending, deletion and update), a selector name, a parameter group (type, value) and an execution code to the object manager **103** through the view manager **101**.

Maruyama, col. 9, lines 5-9.

The user designates an object identifier of a type definition object to the object manager **103** through the view manager **101**.

Maruyama, col. 9, lines 38-40. The cited passages teach that a user designates an object identifier. However, the reference does not teach or suggest “matching the meta definition to graphical user interface fields” or “presenting the graphical user interface fields to the user,” as recited in claim 26. Each and every claim limitation is not taught by the reference; therefore, the claim is not anticipated by *Maruyama*. Claim 31 is allowable for the same reasons.

With respect to claims 25 and 30, the Office Action states:

As per claims 25, 30:

Maruyama teaches means for receiving inputted data values corresponding to the object attributes from the user; and means for sending a data value stream including the inputted data values to a server [e.g., col. 7, lines 60-65, col. 9, line 5, line 39].

Office Action dated 14 December 2000. Applicant respectfully disagrees. The cited passages teach that a user designates an object identifier. However, the reference does not teach or suggest “receiving inputted data values corresponding to the object attributes from the user” and “sending a data value stream including the inputted data values to a server,” as recited in claims 25 and 30. As discussed above, *Maruyama* is concerned only with an object-oriented database and not the manner in which objects are processed in a distributed data processing system. Therefore, *Maruyama* does not teach or suggest the method and data processing system recited in claims 25 and 30 in which a software component queries a Meta Data Service for the meta definition, receives attributes in a soft format (data value stream), and sends the attributes to a server in a soft format.

Therefore, the rejection of claims 1, 2, 5, 7, 10, 13, 14, 16, 18-20, 22, 24-35, 37, 40, 42, and 46 under 35 U.S.C. § 102 is overcome.

Furthermore, *Maruyama* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Absent some teaching or incentive to implement *Maruyama* in a distributed data processing system in which software components pass data objects as data value streams and query a Meta Data Service for meta definitions, one of ordinary skill in the art would not be led to modify *Maruyama* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *Maruyama* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the applicants’ disclosure as a template to make the necessary changes to reach the claimed invention.

III. 35 U.S.C. § 103, Obviousness

The Office Action rejects claims 4, 6, 11, 12, 36, 38, 39, 41, 43-45, and 47 under 35 U.S.C. § 103 as being unpatentable over *Maruyama* in view of well known prior art. This rejection is respectfully traversed.

Since claims 4, 6, 11, 12, 36, 38, 39, 41, 43-45, and 47 depend from claims 1, 7, 35, 42, and 46, the same distinctions between *Maruyama* and the claimed invention in claim 1, 7, 13, 16, 19, 22, and 35 for these claims. Additionally, claims 2, 5, 10, 14, 18, 20, 24-34, 37, and 40 claim other additional combinations of features not suggested by the reference. Therefore, the rejection of claims 4, 6, 11, 12, 36, 38, 39, 41, 43-45, and 47 under 35 U.S.C. § 103 is overcome.

IV. Conclusion

It is respectfully urged that the subject application is patentable over *Maruyama* and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 28 February 2001

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'S. Tkacs', with a long horizontal flourish extending to the right.

Stephen R. Tkacs
Reg. No. 46,430
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001